Christopher Gutierrez

009925875

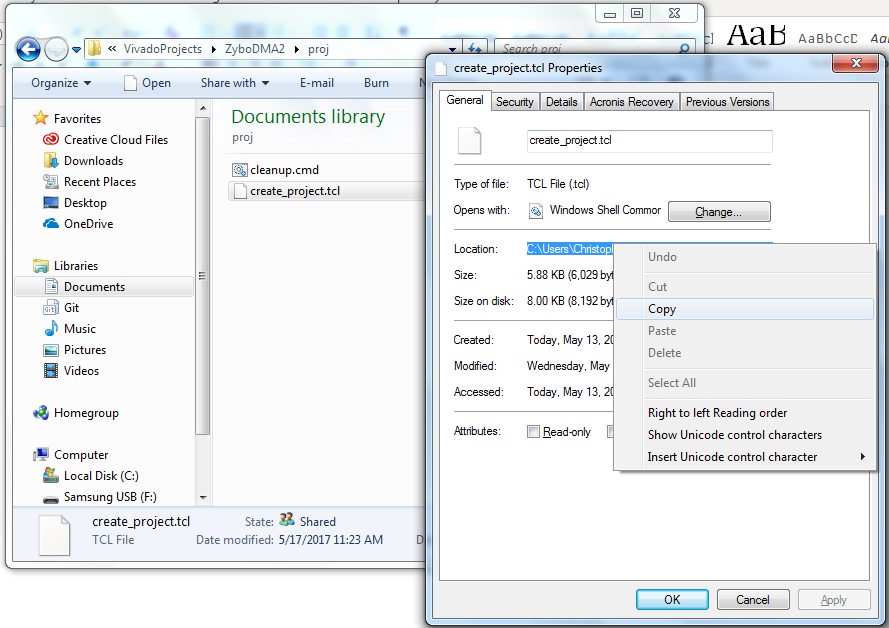
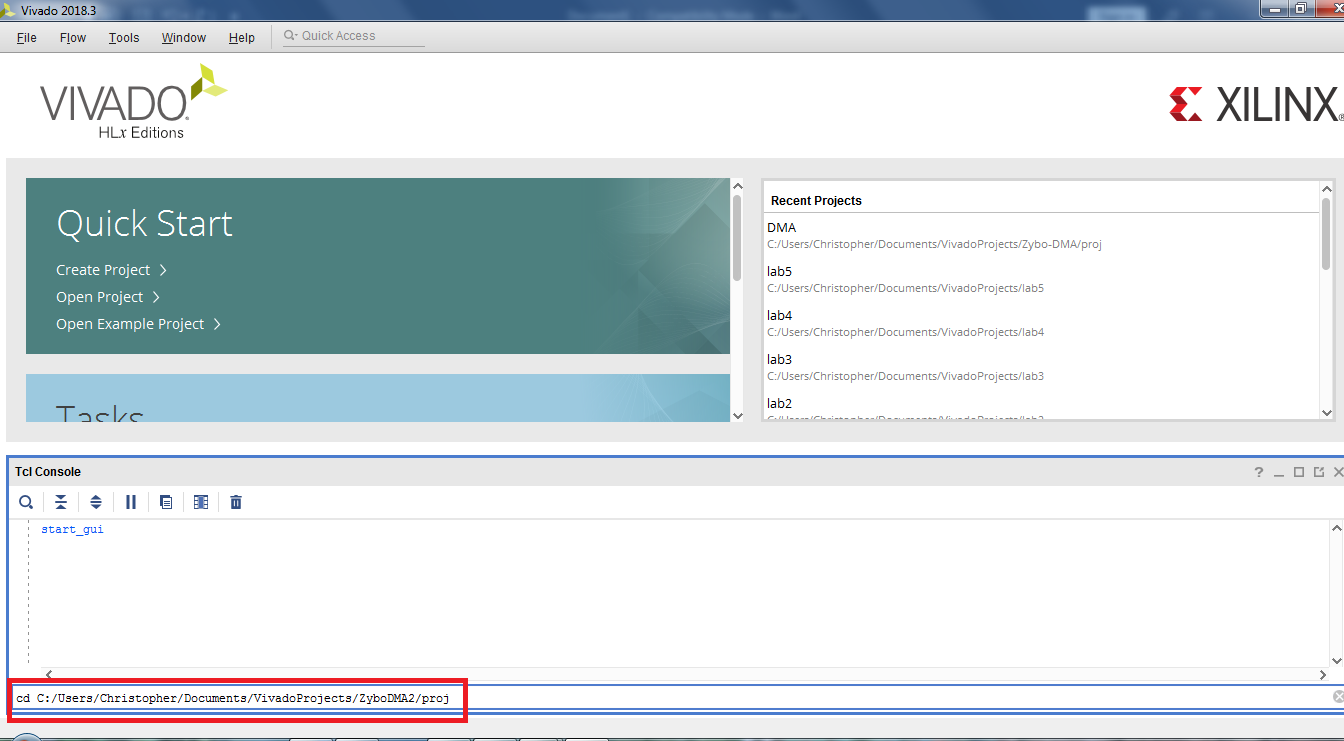
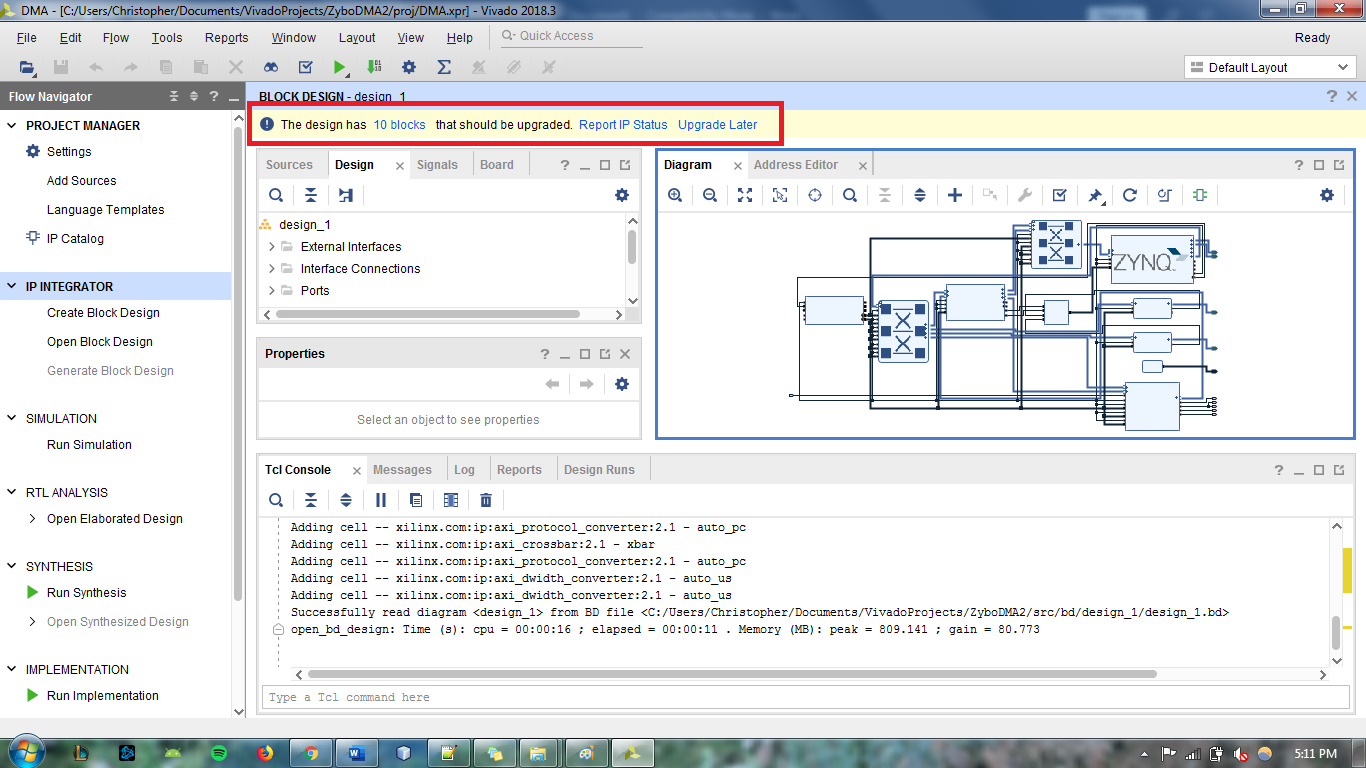
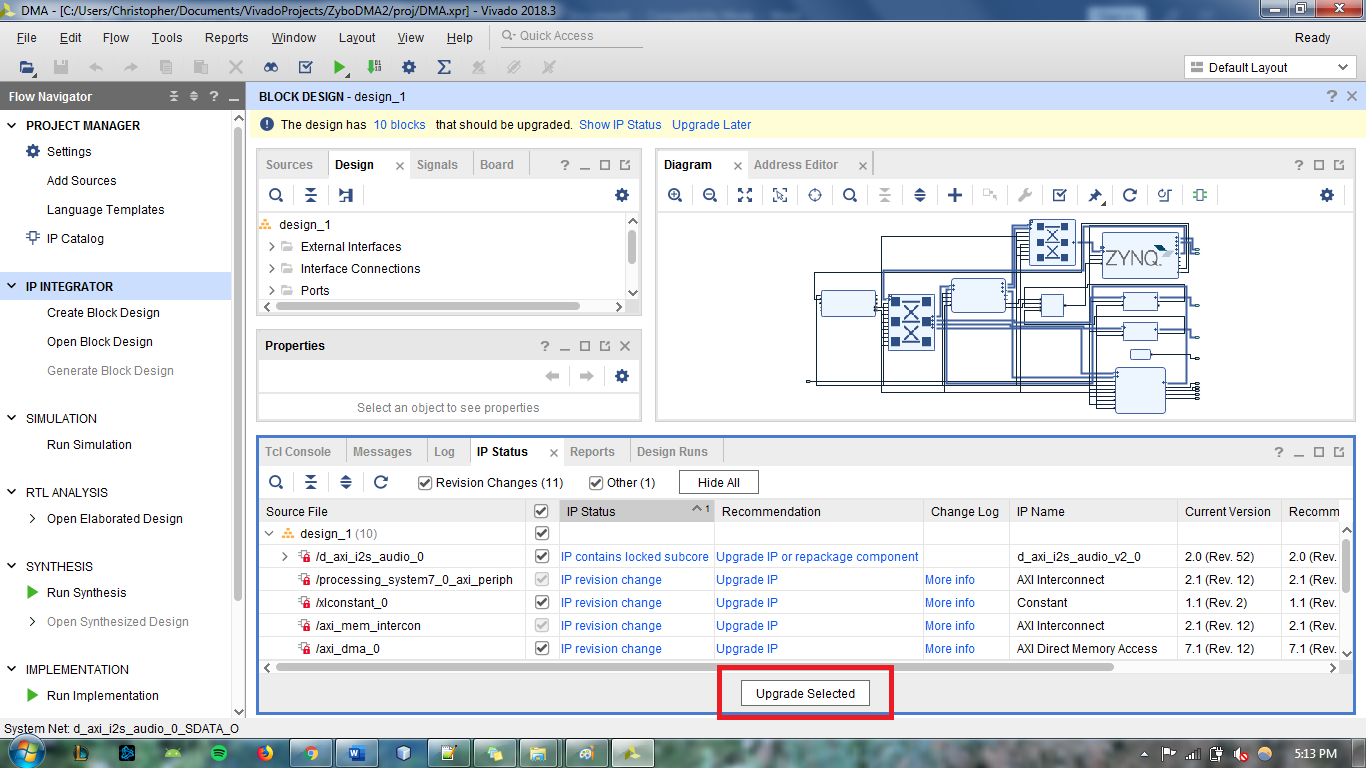
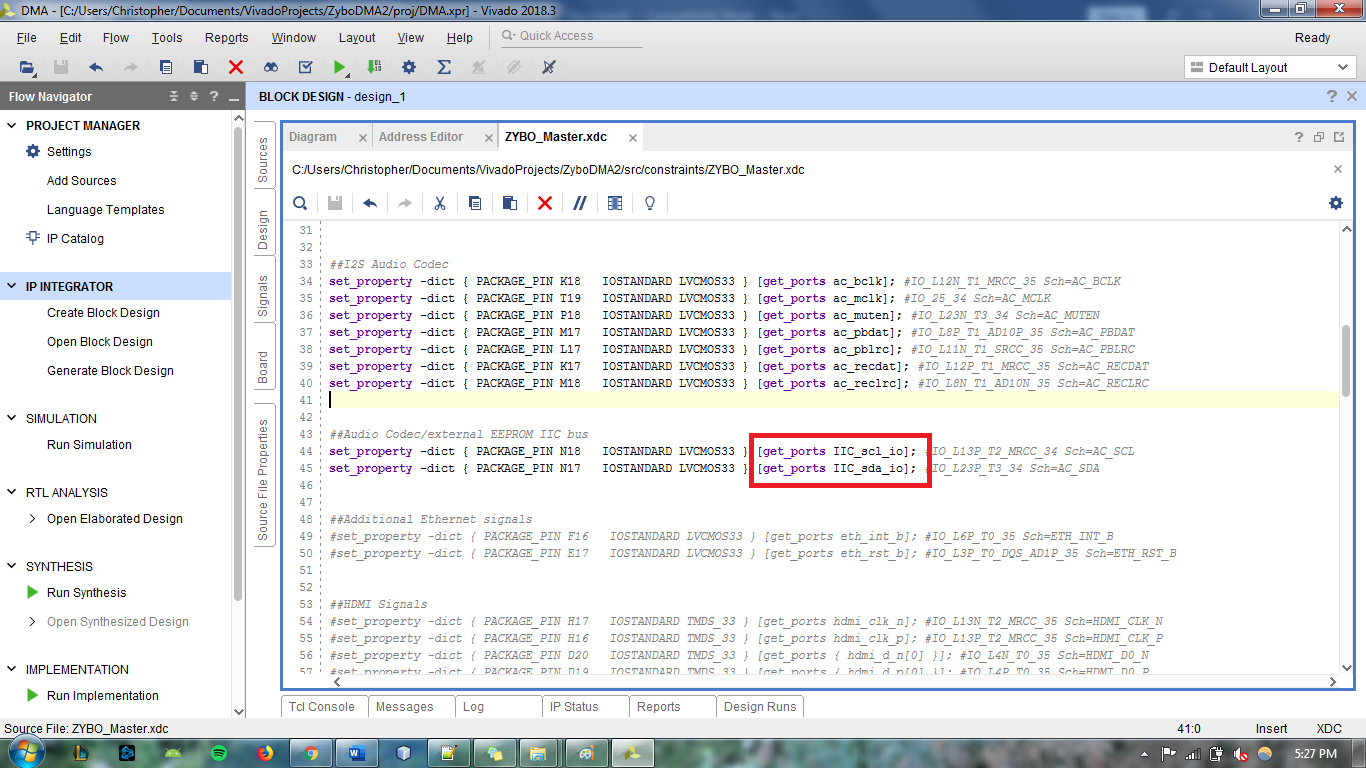
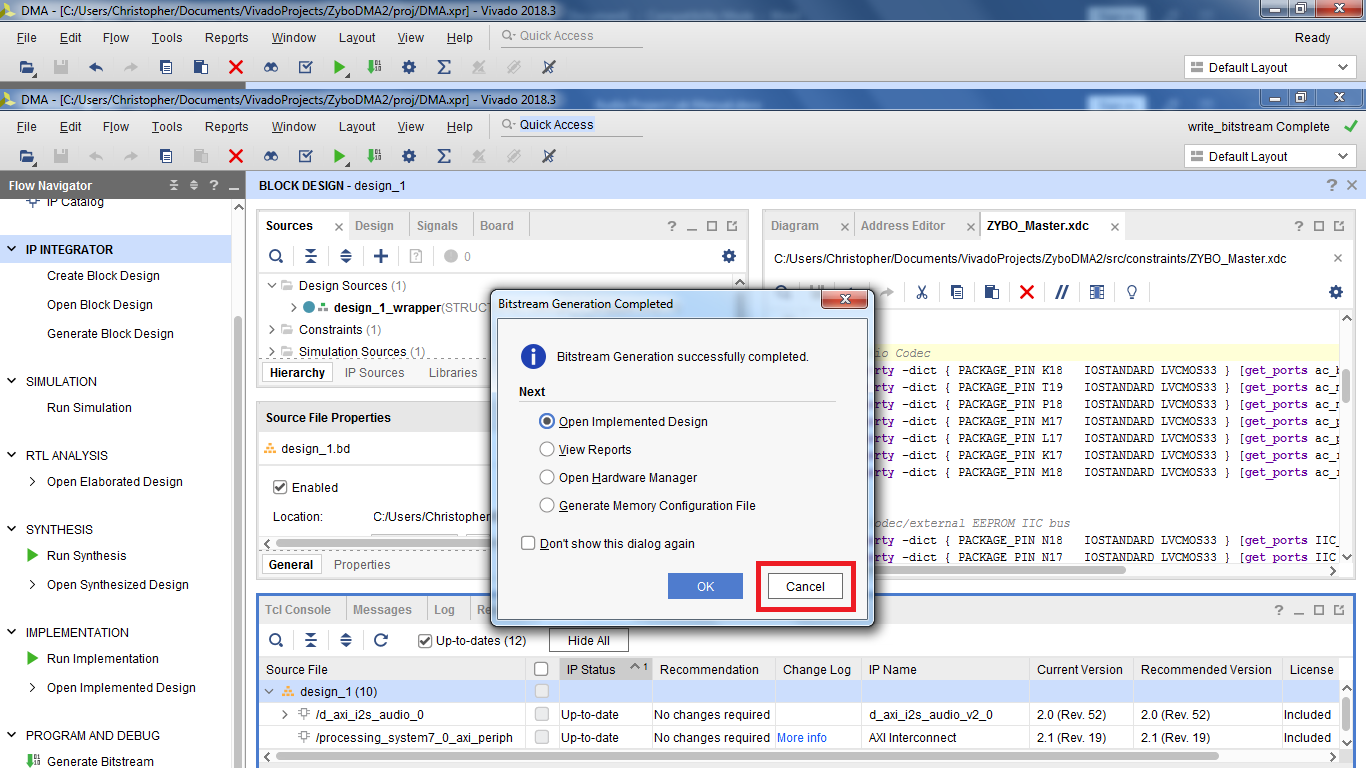
5/13/19

**Final Audio Project Lab Manual**

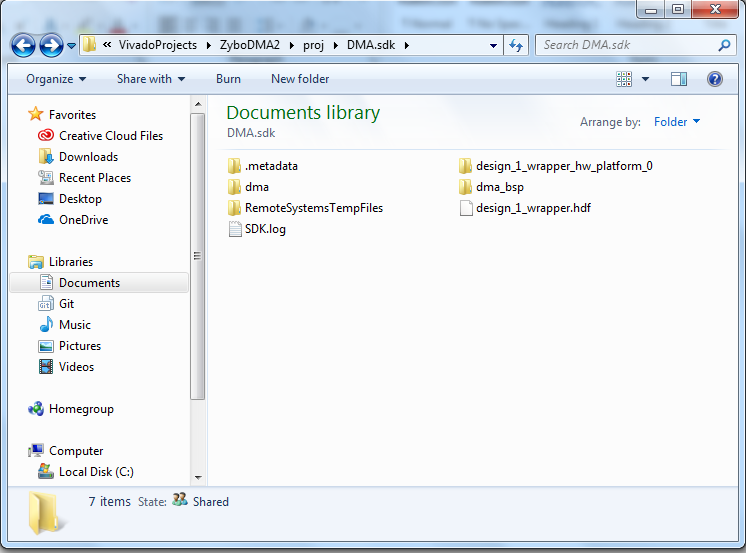
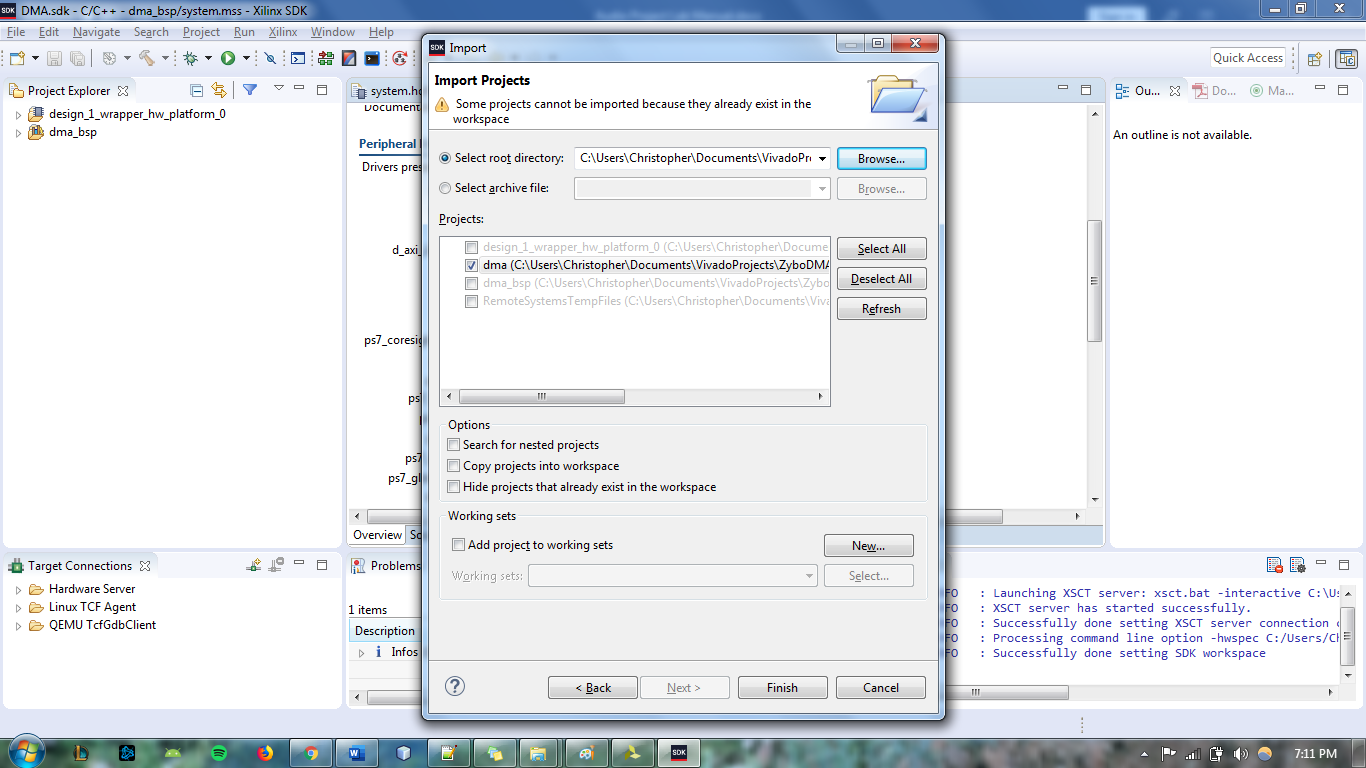
\* Note: For this project we are using Vivado 2018.3, I cannot guarantee this same method will work with any other version of Vivado. \*

**Step 1: Get Digilent’s DMA Audio Demo up and running**

* **Step 1-1: Get the hardware correctly configured**

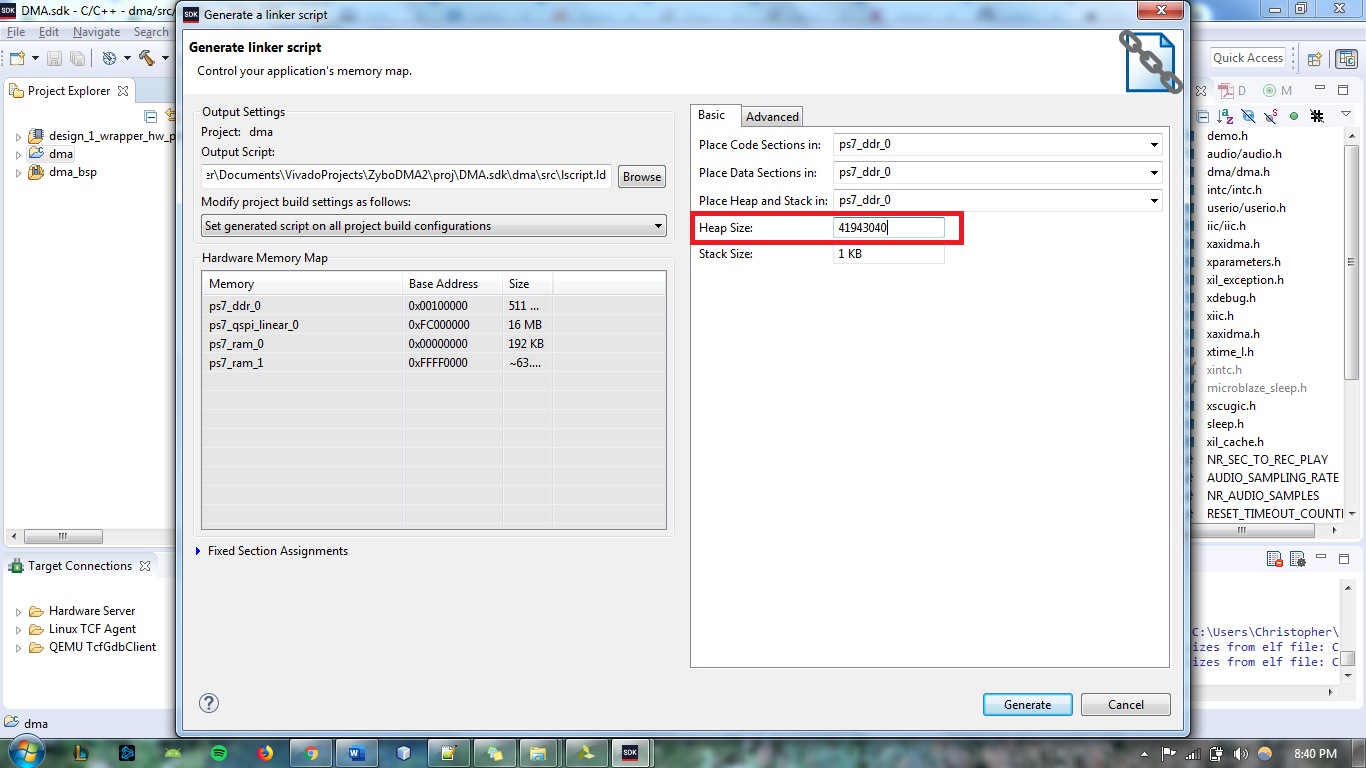
1. **Download** the Digilent Zybo DMA Audio Demo found [here](https://github.com/Digilent/Zybo-DMA/releases/download/v2016.4-3/Zybo-DMA-2016.4-3.zip?_ga=2.164540152.690883232.1557789896-1875058106.1548301546)
2. **Extract** the ZIP project folder into your folder containing your Vivado Projects. For Example, my Vivado Projects are located at “C:\Users\Christopher\Documents\VivadoProjects”
3. Within the project folder there will be several subfolders named “hw\_handoff”, “proj”, “src”, “repo”, etc. Go into the “proj” folder, right click the “create\_project.tcl” file and select properties. **Highlight and copy** this file’s location (we’ll need it in future steps). 
4. **Open Vivado and find** the Tcl Console at the bottom of the window. Note: if you don’t see a console, go to Window > Tcl Console or press CTRL+SHIFT+T.
5. **Enter** the letters “cd ” (change directory) and paste the file path you just copied.
6. Next, **change all the backslashes** ‘\’ in the path into forward slashes ‘/’ and hit enter.
7. **Enter the command** “source ./create\_project.tcl”, this will set up the project for you within the directory you previously cd’d into.
8. You can open the project by double-clicking on the “DMA.xpr” project file in the “proj” folder.
9. Open the block diagram by **clicking on “Open Block Design”** under “Flow Navigator” on the left side of the window.
10. Once the diagram is open, you should get a warning like the following:
11. **Click on “Report IP Status”**
12. Since this project was originally created for an older version of Vivado, the IPs used in it have since changed. Thus, they should be upgraded before we continue. **Click on “Upgrade Selected”** in the “IP Status” tab at the bottom of the screen. 
13. You should see a pop-up window saying the IP Upgrade has completed. **Click ‘OK’.**
14. A ‘Generate Output Products’ window will appear. For now, we will **select ‘Skip’.**
15. Lastly, you will see a ‘Critical Messages’ window. We will ignore these for now. **Click ‘OK’.**
16. Under the same IP Status tab, you will get a warning that the report is out of date. **Click ‘Rerun’** to verify the changes we have made.
17. All the IPs should now be up to date.
18. In the ‘Sources’ tab (top left) **expand the ‘Constraints’ folder** and **double click** on the **‘ZYBO\_Master.xdc’** constraints file to open it.
19. **Scroll down to line 44 and 45. Change the ‘iic\_scl\_io’ port to ‘IIC\_scl\_io’ and change the ‘iic\_sda\_io’ port to ‘IIC\_sda\_io’**. Notice that all we did was capitalize the ‘iic’ in these ports. In order version of Vivado this capitalization was not necessary, but that has since changed in Vivado 2018.3. 
20. Open the block diagram as described previously
21. Add a GPIO IP and configure it to work with the board’s dip switches (not necessary for the demo, but it is easier to do this part now rather than later, resources on how to accomplish this can be found [here](https://reference.digilentinc.com/reference/programmable-logic/zybo/start?redirect=1))
22. Create an HDL Wrapper for our block design by opening the ‘Sources’ tab, expand ‘Design Sources’, right click ‘design\_1’ and select ‘Create HDL Wrapper’
23. Next, in the Flow Navigator panel on the left of the window, scroll down and **click ‘Generate Bitstream’**. Click Yes if prompted to launch Synthesis and Implementation.
24. The number of jobs depends on how many cores your machine has. The more cores, the higher the number of jobs you can have at once. Since my machine only has 2 cores, I will select 2 jobs. If you’re unsure, simply **click ‘OK’**. Note: this step may take quite a while depending on your machine, at this point you might want to get up and stretch or get another cup of coffee!
25. Once the bitstream has been generated, another window will pop-up and say the generation was successful. **Click ‘Cancel’.** 
26. **Click File > Export > Export Hardware**. When the pop-up window comes up, tick the ‘Include bitstream’ box and Click ‘OK’.
27. **Click File > Launch SDK** to launch the SDK and begin working on the software portion of the project.

* **Step 1-2: Create the Board Support Package and Import Application Project**

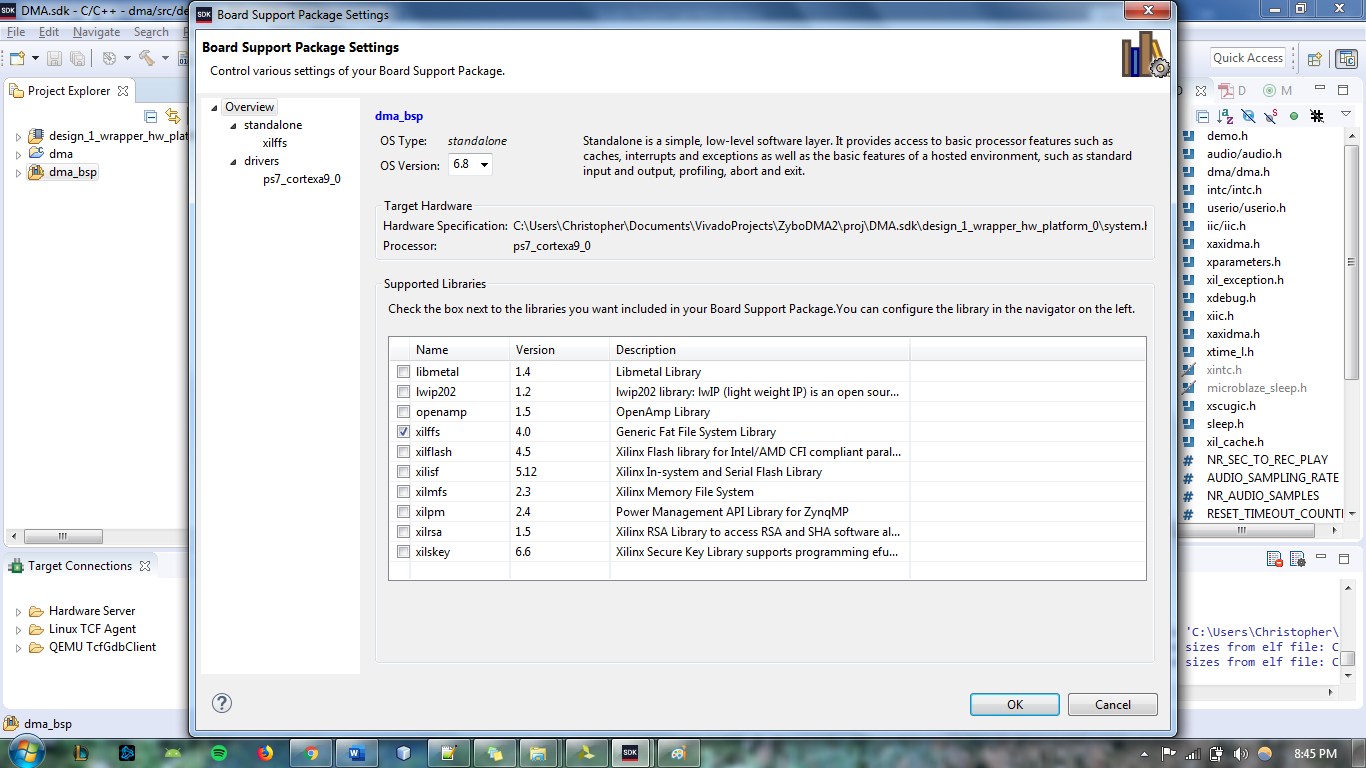
1. In the SDK, go **File > New > Board Support Package.**
2. For the project name, **type in “dma\_bsp”. Click ‘OK’.**
3. **Click ‘OK’** once more.
4. Note that since we updated our hardware in Vivado, we had to generate a new BSP. Thus, we could not use the BSP that was provided to us by Digilent. However, for the application project, we will be using the Digilent provided project. The reason is that the project contains important functions related to the drivers associated with the audio codec.
5. Once again navigate to the folder where your Vivado projects are stored and **open our Zybo-DMA project folder** (the same folder that has “hw\_handoff”, “proj”, “src”, “repo”, etc).
6. Open the ‘sdk’ folder, **right click** the ‘dma’ folder and select **copy**.
7. Go back to the Zybo-DMA project folder and **open the ‘proj’ folder**. Then**, open the ‘DMA.sdk’ folder** and **paste** the previously copied folder into this location.
8. Your ‘DMA.sdk’ folder should now look something like this: 
9. We will now import the application project into our workspace by going to **File > Import.**
10. **Expand the ‘General’ folder** and **select** ‘Existing Projects into Workspace’
11. Under ‘Select Root Directory’ **click on ‘Browse’** and navigate to the ‘DMA.sdk’ folder. Select the folder and **click ‘OK’**. 
12. The ‘dma’ application project should already be selected. **Click ‘Finish’** to import the project.
13. The complete application should now be up and running. Visit the [Audio Demo website](https://reference.digilentinc.com/learn/programmable-logic/tutorials/zybo-dma-audio-demo/start) for instructions on how to use the demo.

**Step 2: Add the Dip Switches GPIO IP, Modify the Application and BSP to complete the audio project**

* **Step 2-1 Modify the application files**

1. For this project, we only modified 4 files (‘audio.h’, ‘audio.c’, ‘demo.h’, ‘demo.c’). However, the modifications to these files are a bit too lengthy to explain each modification one-by-one. Thus, the files can be downloaded from my [GitHub repository](https://github.com/gutierc2/CECS-561-Audio-Project-Files).
2. Replace the files in the application project with the corresponding files found on my GitHub repository (link in previous step).
3. Since our project uses more memory, we must generate a new linker script with more memory added to the heap.
4. In the SDK, right click the dma application project and select ‘Generate Linker Script’.
5. Under the ‘Basic’ tab, enter a heap size of ‘41943040’ and click ‘Generate’. 

* **Step 2-2 Modify the board support package settings**

1. Our project makes use of the microSD IP (enabled in HW by default) and we need to enable the Fat File System library in the board support package settings in order interact with it
2. Right click the ‘dma\_bsp’ board support package application folder in the SDK and select ‘Board Support Package Settings’
3. Under the ‘Supported Libraries’ section, tick the box next to ‘xilffs’ and click ‘OK’ 
4. The project folder should immediately begin updating to include the necessary functions

**How to Use the Audio Project**

\*Note that this project uses the microSD card slot. As such, a microSD card is required for it to work as intended\*

1. Connect the board to your machine via Micro USB cable and turn it on.
2. In the SDK terminal, add the corresponding port.
3. Go to Xilinx > Program FPGA to program the board with our hardware design.
4. Right click the application project and select Run As > Launch on Hardware (System Debugger)
5. This application works much like the demo, with the addition of the DIP Switches. Each combination of the switches will record and playback a different audio track and save it into the microSD card. It is important to note that you **must** record a track before you attempt to play it back. Otherwise, it could cause the application to crash.